ดร.ปัญญา เปรมปรีดิ์

SOFT MAIL

หมากชน ตอนที่ 1

เสนอโปรแกรมเกม "หมากชน" พร้อมทั้งกฎเกณฑ์การเล่น และเค้าโครงของโปรแกรม

อีกเหลืออีก 5 เดือนครึ่งผมก็จะถึงวันปลดเกษียณแล้ว ด้วยเหตุนี้ผมก็เลยหันมาจับเรื่องของการเขียนโปรแกรมให้มากขึ้น การเขียนโปรแกรมนี้ช่วยผมได้ 2 ทาง ทางหนึ่งคือจะช่วยกระตุ้นให้สมองและร่างกายของผมให้ทำงานอย่างสม่ำเสมอ ซึ่งอาจช่วยให้ผมอายุยืนขึ้น ส่วนอีกด้านหนึ่งก็คือ การเขียนโปรแกรมจะทำให้ผมมีเนื้อหาที่พอจะเอามาเขียนเป็นบทความ และจะช่วยเป็นที่มาของรายได้พิเศษ สำหรับครั้งนี้ผมขอเสนอโปรแกรมเกมชนิดใหม่ชนิดหนึ่ง ผมขอเรียกมันว่า "หมากชน" ทั้งนี้เพราะมันมีลักษณะการเดินที่จะต้องวิ่งเข้าหากัน ใครที่หนีไปไหนไม่ได้ก็จะต้องถูกกิน เกมนี้ผมใช้เวลาเขียนอยู่ 10 วัน มันเป็นโปรแกรมที่ยาวพอสมควร ซึ่งถ้าจะนำมาอธิบายกันให้หมดทุกขั้นตอน (เพื่อให้ท่านได้เรียนรู้วิธีออกแบบโปรแกรมและการเขียนโปรแกรม) ก็คงจะยาวมาก ดังนั้นผมจึงต้องขอแบ่งออกเป็น 2 ตอน ตอนที่ 1 นี้จะพูดถึงกฎเกณฑ์การเล่นเกมและเค้าโครงของโปรแกรม ส่วนตอนที่ 2 จะเป็นเรื่องรายละเอียดของโปรแกรม คนที่สนใจในเรื่องเทคนิคการเขียนโปรแกรมก็คงต้องรอไปอีก 1 เดือน

ลักษณะของเกม

เกมนี้เป็นเกมกระดาน นั่นคือ ถ้าเป็นคนรุ่นเก่าก็จะต้องไปหาแผ่นกระดาษขนาดใหญ่มา แล้วต้องตีเส้นเป็นตารางสี่เหลี่ยมจตุรัสวางชิดติดกันเป็นแถวและเป็นคอลัมน์จำนวนหนึ่ง จากนั้นก็ต้องไปเลื่อยไม้หรือแกะสลักไม้ให้เป็นตัวเดินโดยมีขนาดเล็กกว่าช่องตารางบนแผ่นกระดานสักเล็กน้อยมาจำนวนหนึ่ง เมื่อได้แล้วก็มาเริ่มต้นด้วยการวางตัวเดิน (ซึ่งเขาจะเรียกว่า "ตัวหมาก") ลงไปบนช่องต่าง ๆ ให้ครบตามกฎเกณฑ์ เกมประเภทนี้จะมีผู้เล่น 2 คน ซึ่งจะนั่งอยู่ตรงข้ามกัน และวางแผ่นกระดานอยู่ตรงกลาง ผู้เล่นจะผลัดกันเดิน เขาจะเดินได้ครั้งละ 1 ตัว ก่อนจะเริ่มเดินก็ต้องมีการเสี่ยงทายกันว่าใครจะได้เดินก่อน คนโบราณส่วนใหญ่จะใช้วิธีเอาเหรียญออกมาโยนขึ้นไปสูง ๆ และให้ตกลงมาบนพื้นโดยไม่มีการเข้าไปขัดขวางหรือสอดแทรก เมื่อเหรียญตกลงมาแล้วก็ดูว่าด้านไหนหงายขึ้นมา ถ้าออกด้านหัวให้ฝ่ายแดงเดินก่อน ถ้าออกด้านก้อยก็ให้ฝ่ายดำเดินก่อน แต่มาถึงยุคของคอมพิวเตอร์เราก็จะเปลี่ยนมาเป็นการสร้างภาพบนจอภาพ ทั้งนี้โดยให้รูปภาพนั้นมองเหมือนเป็นกระดาน และมีตัวหมากวางอยู่บนกระดานตามกฎเกณฑ์ที่เรากำหนดขึ้น การหยิบตัวหมากเพื่อเดินไปมาก็ถูกเปลี่ยนมาเป็นการเคาะแป้นคีย์บอร์ด หรือไม่ก็ใช้เม้าส์ลากไปมา นอกจากนี้แล้วเราก็มักจะปรับเปลี่ยนมาเป็นว่าท่านเป็นผู้เล่นด้านหนึ่ง ส่วนอีกด้านหนึ่งจะคิดและเดินโดยเจ้าเครื่องคอมพิวเตอร์ หรือพูดง่าย ๆ ก็คือเปลี่ยนจากการเอาคนมาเล่นแข่งกันมาเป็นคนเล่นแข่งกับคอมพิวเตอร์ **ความสนุกมันอยู่ตรงที่ว่า ท่านจะเอาชนะคนเขียนโปรแกรมได้หรือไม่ และถ้าท่านมีเครื่อง 2 เครื่อ**

ท่านก็อาจให้คนเขียนโปรแกรม 2 คนไปเขียนโปรแกรมเพื่อเอามาแข่งขันกันได้ สำหรับเกม "หมากชน" อันนี้ผมออกแบบให้เหมาะสำหรับการส่งเสริมให้มีการเขียนโปรแกรมมาแข่งกัน ที่ผมบอกว่ามันเหมาะก็เพราะกฎเกณฑ์ในการเล่นเกมนี้จะเป็นกฎเกณฑ์ง่าย ๆ มันซับซ้อนน้อยกว่าหมากรุกและหมากฮอสหลายเท่าตัว แต่การวางหลักการเพื่อเอาชนะคู่ต่อสู้นั้นอาจต้องใช้เวลาคิดและทดลองนานพอสมควร มันเหมาะสำหรับการฝึกสมองของพวกเรา (นักคอมพิวเตอร์) มากทีเดียว

กฎเกณฑ์การเล่น

ตัวหมากสำหรับเล่นในเกมนี้จะมีลักษณะเป็นเหรียญกลม ๆ ด้านหนึ่งจะบอกสีและอายุ ส่วนอีกด้านหนึ่งจะไม่มีอะไรบอกเลย ด้านแรกนั้นเรียกว่า "ด้านหน้า" ส่วนด้านที่ 2 นั้นเรียกว่า "ด้านหลัง" ผู้เล่นแต่ละคนจะมีตัวหมากคนละ 16 ตัว ในตอนเริ่มต้นนั้นเราจะให้เอาหมากทั้งหมด (= 32 ตัว) คว่ำหน้าลง และเนื่องจากด้านหลังของตัวหมากทุกตัวไม่มีสีและไม่มีอะไรบอกเลยมันจึงดูแล้วเหมือนกันหมด ผู้เล่นจะต้องใช้มือกวาดตัวหมากให้ผสมกันจนทั่ว แล้วแยกกระจายใส่ไปตามช่องต่าง ๆ อย่างทั่วถึง สำหรับกระดานที่จะใช้เล่นเกมนี้นั้นจะแบ่งเป็น 8 แถว แถวละ 8 ช่อง หรือคือเป็นช่อง 4 เหลี่ยมจตุรัสรวม 64 ช่อง ซึ่งเหมือนกับตารางหมากรุกไทยนั่นแหละ

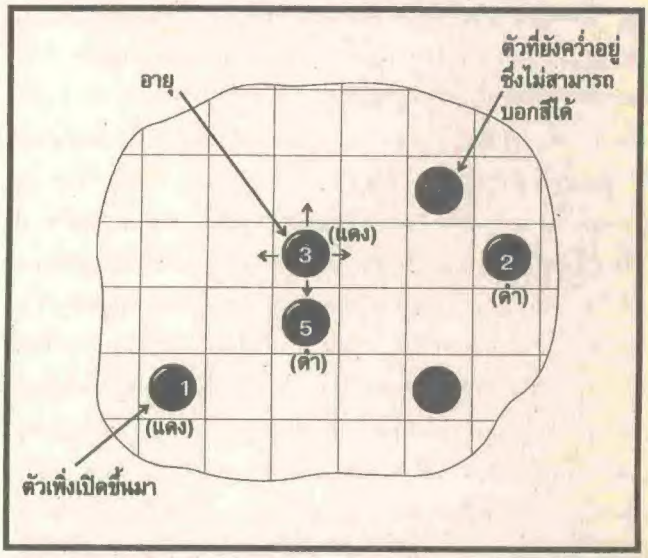
หลังจากที่มีการโยนเหรียญหรือจับฉลากกันแล้วก็จะมีการเล่น ผู้เล่นนั้นโดยทั่ว ๆ ไปจะมีทางเลือก 2 ทาง คือ

(ก) เดินตัวใดตัวหนึ่งที่หงายหน้าอยู่แล้ว และเป็นสีของของฝ่ายตน

(ข) ขอเปิดตัวใดตัวหนึ่งที่ยังคว่ำหน้าอยู่

แต่ผู้เล่นรายแรกและเล่นเป็นครั้งแรกนั้นจะทำได้แต่ข้อ (ข) เท่านั้น ทั้งนี้เพราะในตอนเริ่มต้นนั้นหมากทุกตัวอยู่ในสถานะคว่ำหน้าอยู่ เมื่อใดที่มีการเปิดตัวใหม่ขึ้นมาเราก็จะดูว่าเป็นสีของฝ่ายใด แล้วเราก็เขียนตัวเลข "1" ลงไปบนหน้าของหมากตัวนั้น เจ้าตัวเลขอันนี้คือตัวเลขบอกอายุของตัวหมาก ในกรณีของการเดินนั้นเราจะเดินได้แค่ 4 ทิศ คือขึ้นข้างบนหรือลงข้างล่าง ไปทางซ้ายหรือไปทางขวา และเป็นระยะทางแค่ 1 ช่องเท่านั้น การเดินไปนี้จะเดินไปได้ก็เมื่อช่องที่จะเดินไปถึงนั้นเป็นช่องที่ว่างเปล่า หรือเป็นช่องที่มีตัวที่หงายแล้วและเป็นของฝ่ายตรงข้ามวางอยู่ ในกรณีหลังนี้เราเรียกว่าการ "กิน" นั่นคือเราจะหยิบเอาตัวหมากของฝ่ายตรงข้ามออกไป แล้วเอาตัวหมากของเราเข้าไปวางแทน ในการเดินแต่ละครั้งก็จะทำให้อายุของตัวหมากเพิ่มขึ้นอีก 1 หน่วย นั่นคือ ท่านจะต้องลบอายุเดิมออกแล้วเขียนอายุใหม่ลงไปแทน ตัวหมากเหล่านี้จะมีอายุได้สูงสุด 9 หน่วย หรือคือ ตัวหมากที่มีอายุถึง 9 หน่วยแล้วนั้นจะเดินได้อีกครั้งเดียว เมื่อเดินแล้วก็จะถูกยกออก ทั้งนี้เพราะ "หมดอายุขัย"

เป็นอย่างไรครับ กฎเกณฑ์ในการวางหมากและการเล่นนั้นไม่ยากเลย เราสามารถเขียนโปรแกรมเพื่อวาดรูปตารางหมากชนและหยอดตัวหมากลงไปได้โดยง่าย การมารับคำสั่งเพื่อเดินหรือเปิดตัวหมากนั้นก็ไม่ยาก ท่านจะใช้วิธีคีย์เป็นรหัสข้อความเข้าไปก็ได้ เช่น คีย์เข้าไปว่า "DERN1516" ก็จะหมายถึงการเดินตัวหมากที่อยู่ในช่องที่ (1,5) ไปยังช่องที่ (1,6) ส่วน "PERD37" ก็อาจหมายถึงการสั่งให้เปิดตัวหมากที่ช่อง (3,7) ก็ได้ สำหรับของผมนั้นได้อาศัยโปรแกรมเล่นหมากรุกไทยที่เขียนเอาไว้เมื่อ 10 ปีก่อนโน้นมาเป็นจุดเริ่มต้น ผมตัดทอนเอาส่วนอื่น ๆ ทิ้งไป ลดเหลือแค่การกดเครื่องหมายลูกศรชี้ 4 ทิศ และการกดแป้น ENTER เพื่อการเลื่อนจุดเป้าหมายและการสั่งดำเนินการเอาไว้ ผมก็สามารถได้ตัวโครงสร้างเพื่อการวาดภาพและการรับคำสั่งโดยทันที ส่วนที่เหลือก็เป็นเรื่องของลอจิกการเดินหมากของฝ่ายคอมพิวเตอร์ แต่อย่างไรก็ตาม มันก็ทำให้ผมนอนดึกไป 10 คืนเลยทีเดียว



รูปที่ 1 แสดงตัวอย่างการเดินของตัวฝ่ายแดง

ขอโทษด้วยครับ พูดมาตั้งนาน แต่ยังไม่ได้บอกเลยว่าเกมนี้จะแพ้หรือชนะกันอย่างไร การแพ้หรือชนะนั้นมี 2 รูปแบบ และเกมนี้จะไม่มีการเสมอกัน สาเหตุก็เพราะการแพ้หรือชนะกันในรูปแบบที่ 1 คือ (ค) **ใครหมดหนทางเล่น (เดินหรือเปิด) ก่อนจะเป็นผู้แพ้** เจ้ากฎเกณฑ์ข้อ (ก), (ข) บังคับว่าเราจะต้องเดินหรือไม่ก็เปิดตัวที่ยังคว่ำหน้าอยู่ ซึ่งทั้ง 2 อย่างนี้จะนำมาซึ่งการหมดไปของตัวเล่น เช่น เดินไปแต่ละครั้งก็จะทำให้อายุของตัวหมากเพิ่มขึ้น และเมื่อเดินไปเป็นครั้งที่ 9 ก็จะตายไป ส่วนการเปิดนั้นก็จะทำให้ทางเลือกข้อ (ข) ค่อย ๆ ลดลง เปิดไปไม่นานก็ต้องหงายหน้าจนหมด ซึ่งหลังจากนั้นแล้วทุกตัวก็จะเดินไปสู่หลุมศพเช่นเดียวกับมนุษย์เรา ทุกคนเกิดมาแล้วก็ต้องตาย แต่หมากชนอันนี้มันไม่มีการตกไข่เพิ่มขึ้นได้ ลงท้ายมันก็ต้องหมดไปจากกระดาน

สำหรับการแพ้ชนะกันในรูปแบบที่ 2 คือ (ง) **ในการเปิดและการกินนั้นจะมีการให้คะแนน และเมื่อคะแนนของใครมากกว่าฝ่ายตรง**

ข้ามจนถึงขีดจำกัดอันหนึ่ง (score limit) ฝ่ายนั้นก็จะชนะไปเลย คะแนนในการเปิดและการกินจะเป็นดังนี้

(ง.1) เปิดออกมาแล้วพบว่าเป็นหมากของฝ่ายตนจะได้คะแนน 1 แต้ม

(ง.2) เปิดออกมาแล้วพบว่าเป็นหมากของฝ่ายตรงข้ามจะเสียคะแนน 2 แต้ม

(ง.3) กินฝ่ายตรงข้ามจะได้คะแนน 3 แต้ม

กลยุทธ์ในการเล่น

ดูจากกฎเกณฑ์ในเรื่องการแพ้ชนะกันนี้แล้วจะเห็นได้ว่า การวางกลยุทธ์ (strategy) เพื่อเอาชนะนั้นจะยุ่งยากพอสมควร กฎข้อ (ค) นั้นจะทำให้เราต้องหาทางกินฝ่ายตรงข้ามให้ได้มากที่สุดเท่าที่จะมากได้ กฎเกณฑ์ข้อ (ง.3) จะยิ่งช่วยย้ำแนวทางดังกล่าว แต่กฎเกณฑ์ข้อที่ (ง.1) และ (ง.2) นั้นมันบอกว่า **เราไม่ควรเป็นผู้เปิดตัวหมาก** ทั้งนี้เพราะเปิดผิดจะเสียหายได้มากกว่าการเปิดถูก แต่อย่างไรก็ตาม ถ้าเราจดบันทึกไว้ดี ๆ เราก็อาจบอกได้ว่าตอนไหนเรามีความเสี่ยงอยู่เท่าใด โดยเฉพาะอย่างยิ่งเมื่อถึงตัวสุดท้ายนั้นท่านน่าจะบอกได้เลยว่าจะเป็นสีอะไร ดังนั้นกฎเกณฑ์ข้อ (ง.1) และ (ง.2) นี้มันจะส่งผลต่อการตัดสินใจที่กลับไปกลับมาได้ เรื่องนี้ท่านจะต้องนำกลับไปคิดให้ดี ๆ แล้วท่านจึงจะลงมือเขียนโปรแกรมในส่วนที่เป็นโลจิกของการเดินของคอมพิวเตอร์ได้ ถ้ายังวางหลักการหรือกลยุทธ์ไม่ถูก โปรแกรมก็จะเขียนไม่ได้ อีกสิ่งหนึ่งที่จะทำให้การวางกลยุทธ์นั้นยุ่งยากขึ้นก็คือ ตัว "score limit" นั้นมันเปลี่ยนแปลงได้ ค่านี้ผมกำหนดให้ผู้เล่นต้องใส่เข้าไปในตอนเริ่มต้นของเกม จากการทดลองเล่นกับเครื่องประมาณ 20 ครั้ง ผมพบว่าถ้า "score limit" อยู่ต่ำกว่า 20 เกมก็มักจะจบลงด้วยกฎข้อ (ง) แต่ถ้าใส่เข้าไปสูงกว่า 20 เกมก็มักจะจบลงด้วยกฎข้อ (ค) หรือคือเปิดกันจนหมดทุกตัว แล้วก็ถูกกินจนหมดไปข้างหนึ่ง แต่อย่างไรก็ตาม เราจะมาเขียนโปรแกรมโดยมีหลาย ๆ ลอจิกให้เลือกนั้นก็เป็นเรื่องยุ่งยาก ผมจึงต้องใช้วิธีคาดเดาไปก่อน และวางลอจิกตามการคาดเดาดังกล่าว

ผมคิดและวางกลยุทธ์อย่างไร?

นั่นเป็นสิ่งที่น่าจะเอามาถกเถียงกันเล่น มันจะช่วยเสริมทักษะในการออกแบบระบบงานและระบบโปรแกรม

สิ่งแรกที่ผมอยากจะชี้ให้ท่านคิดก็คือ ในการที่จะเข้าไปกินตัวหมากของฝ่ายตรงข้ามนั้นเป็นเรื่องที่เป็นไปไม่ได้ เรื่องนี้ท่านจะต้องทำความเข้าใจให้ดี ๆ "หมากชน" นี้ไม่เหมือนกับ "หมากรุก" หรือ "หมากฮอส" ทั้งนี้เพราะหมากชนเดินได้แค่ 1 ช่องเท่านั้น ถ้าท่านเดินตัวของท่านเข้าไปประชิดตัวของฝ่ายตรงข้าม ฝ่ายเขาก็จะกินฝ่ายเราเสียก่อน เรื่องนี้ท่านทดลองดูได้ คือลองเอากระดาษมา 1 แผ่น แล้วตีเป็นตารางสัก 5 x 5 ช่อง เสร็จแล้วเอาเหรียญ 5 บาท กับ 10 บาท มาอย่างละ 2 - 3 เหรียญ แล้ววางเหรียญเหล่านี้ลงไป ท่านลองเดินทีละข้าง แล้วท่านก็จะพบว่า การเดินเข้าไปเพื่อประชิดตัวและหวังจะกินฝ่ายตรงข้าม

นั้นทำไม่ได้ แต่.. หลังจากที่ท่านทดลองเดินดูสักพักหนึ่งแล้วก็จะเกิดความคิดใหม่ กลยุทธ์ในเรื่องนี้ก็คือ **ต้องตะล่อมฝ่ายตรงข้ามให้เข้าสู่มุมอับ แล้วเขาก็จะต้องเดินออกมาให้เรากิน หรือไม่ก็ต้องเปิดตัวใหม่ ซึ่งเขาจะมีโอกาสเสียมากกว่าได้ (ตามกฎเกณฑ์ข้อ ง.1 และ ง.2)**

เราได้กลยุทธ์ใหม่ที่ดีกว่าเก่า แต่ก็จะมีคำถามว่ามันจะแปลออกมาเป็นลอจิกของโปรแกรมได้อย่างไร สิ่งนี้เป็นสิ่งที่ต้องมองไปข้างหน้าหลายสเตป เมื่อกระดานมีขนาดถึง 8 x 8 และมีตัวที่หงายแล้วข้างละเป็นสิบ กว่าจะบีบไปจนเขาเดินไม่ได้นั้นมันอาจต้องมองลึกไป 5 ถึง 10 ชั้น เจ้าเครื่องคอมพิวเตอร์ของเราจะใช้เวลาเป็นปี ๆ จึงจะมองได้ลึกขนาดนั้น เรื่องนี้คงจะต้องทิ้งเอาไว้ก่อน เมื่อไหร่ที่มีเวลาก็อาจนำมาคิดและตัดทอนดู

ยังครับ ยังไม่ถึงกับตีบตันไปเสียหมด การคิดแบบข้างต้นนั้นเป็นแนวคิดแบบ "เชิงรุก" (Offensive) ซึ่งถ้าคิดได้และทำออกมาเป็นโปรแกรมได้ก็จะมีแต่ชนะกับชนะ แต่นั่นก็ไม่ใช่หนทางเดียวที่มีอยู่ หนทางที่ 2 คือ คิดแบบ "เชิงรับ" (Defensive) หรือคือ ถามตัวเราเองว่าเดินอย่างไร หรือเปิดอย่างไรจึงจะไม่เสียหาย และถ้ามันจะต้องเกิดความเสียหายก็ควรจะเลือกอันที่เสียหายน้อยที่สุด เรื่องนี้ดูแล้วไม่ยาก เราควรแบ่งแยกเป็น 2 กรณี คือการเดินกับการเปิดตัวใหม่ ในเรื่องของการเดินนั้นเราต้องให้เครื่องคำนวณหาให้หมดว่าเดินได้ที่หนทาง เสร็จแล้วเรามาลองจำลองภาพดู คือก๊อบปี้สภาพของกระดานมา แล้วลองเดินตามหนทางที่ 1 ไป แล้วคำนวณว่าฝ่ายตรงข้ามจะเดินได้กี่หนทาง ในบรรดาการเดินของฝ่ายตรงข้ามเหล่านั้นเขากินเราได้บ้างไหม หรือคือให้คำนวณการสูญเสียสูงสุดของฝ่ายเรา แล้วเรียกค่าอันนี้ว่า LOST (ในโปรแกรมผมเรียกมันว่า CLST ตัวอักษร C ที่นำหน้านั้นใช้กับตัวแปรทางด้านคอมพิวเตอร์ ส่วน LST นั้นก็คือคำย่อของคำว่า LOST) มันจะเป็นการสูญเสีย... ถ้าเราเดินไปตามหนทางที่ 1 เมื่อได้แล้วก็วนไปคำนวณให้การทดลองเดินตามทางเลือกที่ 2, 3,... ไปจนครบหมดทุกหนทาง หลังจากได้ค่า CLST จนครบแล้วเราก็สามารถจะดูได้ว่าทางเลือกอันไหน

ขาดทุน อันไหนเท่าทุน และอันไหนมีกำไร จากตัวเลขเหล่านี้เราก็สามารถจะเลือกอันที่ดีที่สุดได้

สำหรับกรณีของการเปิดนั้นก็เช่นเดียวกัน คือเราสามารถคำนวณได้ว่ามีตัวที่คว่ำอยู่กี่ตัว แต่ละตัวก็จะเป็นหนทางเลือก 1 หนทาง แต่การคำนวณค่า CLST นั้นยากกว่าหน่อยหนึ่ง คือในเกมจริง ๆ นั้นผู้เล่นทั้ง 2 ฝ่ายต่างจะไม่รู้ว่าตัวคว่ำเหล่านี้จะเป็นสีอะไร ดังนั้นคอมพิวเตอร์จะต้องทำตัวให้เหมือนผู้เล่นธรรมดา คือต้องไม่แอบไปมองดูในข้อมูลว่าเป็นสีอะไร คอมพิวเตอร์จะต้องสมมติให้เป็น 2 กรณี คือถ้าเปิดออกมาเป็นสีของตนเองจะเป็นอย่างไร และถ้าเปิดออกมาแล้วเป็นของฝ่ายตรงข้ามจะเป็นอย่างไร 2 กรณีนี้ผมสร้างเป็นโพรซีเดอร์ไว้ในชื่อว่า PERDDAI และ PERDSIA แต่ละกรณีเราสามารถคำนวณค่า CLST ได้ ซึ่งปกติแล้วก็จะมีค่าไม่เท่ากัน จากค่าทั้ง 2 นี้ผมเอามาบวกกันแล้วหารด้วย 2 หรือคือเอามาเฉลี่ยกันดื้อ ๆ ไปเลย เรื่องนี้จริง ๆ แล้วเราควรใช้วิธีให้เครื่องจดบันทึกเอาไว้ว่าตัวหมากของฝ่ายใดถูกเปิดไปเป็นจำนวนเท่าใดแล้ว และจะมีเหลืออยู่อีกฝ่ายละกี่ตัว จากตัวเลขนี้เราสามารถบอกได้ว่า โอกาสที่จะเปิดได้เป็นตัวของฝ่ายตนเองนั้นเป็นเท่าใด (= $n1/(n_1+n_2)$ เมื่อ n_1 = ตัวที่เหลืออยู่ของฝ่ายตน n_2 = ตัวที่เหลืออยู่ของฝ่ายตรงข้าม) แต่เพื่อให้เราได้โปรแกรมออกมาทดลองเล่นโดยเร็ว ผมก็ใช้หลัก 50 : 50 ไปก่อน

โดยสรุปแล้วเราสามารถคำนวณจำนวนหนทางเลือกของผู้เล่นได้ และในแต่ละหนทางเลือกนั้นเราก็สามารถคำนวณ "ค่าได้กิน" และ "ค่าถูกกิน" ได้ การคำนวณหนทางเลือกทั้งหมดนั้นผมจับเอามารวมกันเป็นโพรซีเดอร์ (Procedure) อันหนึ่งและใช้ชื่อว่า GENMOVES (...) โพรซีเดอร์นี้จะมีพารามิเตอร์ตามมาหลายตัว และผมแนะนำว่าควรเขียนให้มันใช้ได้กับทั้ง 2 ฝ่าย คือถ้าเอาตัวแปรทางด้านคอมพิวเตอร์ยัดเข้าไป มันก็จะส่งผลออกมาเป็นหนทางเดินของฝ่ายคอมพิวเตอร์ แต่ถ้าเราเอาตัวแปรทางด้านฝ่ายตรงข้าม (เรียกว่า OPPSIDE) ใส่เข้าไป มันก็จะส่งผลออกมาเป็นหนทางเดินของฝ่ายตรงข้าม โพรซีเดอร์แบบนี้สามารถนำมาใช้ซ้ำ ๆ ในระดับต่าง ๆ ได้โดยไม่มีขีดจำกัด นั่นคือ เราสามารถแตกเป็นกิ่งก้านของหนทางเลือกกี่ชั้นก็ได้ ข้อจำกัดจะไปอยู่ที่เวลาในการคำนวณ

เมื่อเราได้ค่าของการ "ได้กิน" และ "ถูกกิน" มาแล้วเราก็สามารถจัดแยกทางเลือกต่าง ๆ ออกเป็นชนิดต่าง ๆ ได้ ถ้าจะคิดแบบง่าย ๆ ก็อาจแบ่งเป็นแค่ 3 ชนิด คือ กำไร เท่าทุน และขาดทุน แต่ผมเสนอแนะว่าควรแบ่งเป็นกลุ่มย่อยลงไปอีกเล็กน้อย ทั้งนี้เพราะในหมู่ที่มีกำไรเท่ากัน หรือเท่าทุนเหมือนกัน หรือขาดทุนเท่ากันนั้น มันยังมีข้อดีข้อเสียที่แตกต่างกันอยู่บ้าง เช่น เท่าทุนเพราะได้กินมา 1 ตัวและเสียไป 1 ตัวนั้นน่าจะดีกว่าการเดินไปฟรี ๆ ที่ผมคิดอย่างนี้เพราะการได้กินนั้นเป็นสิ่งที่เกิดขึ้นแน่นอน ส่วนค่า CLST นั้นเป็นการคาดเดาว่าฝ่ายตรงข้ามจะกินกลับเข้ามา แต่เอาจริง ๆ แล้วเขาอาจเดินหนีก็เป็นได้ นอกจากนี้แล้ว การเล่นแบบ "ตั้งรับ" นั้นเราควรทำให้ตัวเดินลดลงให้เร็วที่สุด เรื่องนี้ผมอาจคิดผิดก็เป็นได้ แต่มันก็ไม่น่าจะเป็นข้อเสียที่ร้ายแรงอะไร

โครงสร้างของโปรแกรม

อันที่จริงแล้วเรื่องของกลยุทธ์นั้นยังมีเรื่องที่จะต้องอธิบายอีกหลายอย่าง แต่เพื่อไม่ให้ท่านผู้อ่านเกิดอาการเมาหมัด ผมจึงใคร่ขอผ่านไปถึงเรื่องโครงสร้างของโปรแกรมเสียก่อน โครงสร้างพื้นฐานเพื่อให้เราสามารถโต้ตอบกับเครื่องได้นั้นจะเป็นดังนี้

(1) จัดการวาดภาพเป็นกระดานหมากรุกขึ้นมา 1 กระดาน

(2) จัดการหยอดตัวหมากของฝ่ายทั้ง 2 ลงไปอย่าง Randomly ฝ่ายละ 16 ตัว โดยให้อยู่ในสถานะคว่ำอยู่

(3) วิ่งมารับค่าของ score-limit

(4) เซตให้ฝ่ายคนเป็นผู้เล่นก่อน

(5) วนอยู่ในลูปเพื่อรับคำสั่ง

(6) เมื่อมีคำสั่งเข้ามาก็ให้ตรวจว่าถูกต้องตามกฎเกณฑ์หรือไม่ ถ้าผิดกฎเกณฑ์ให้ทำเสียง "Beep" แล้วกลับไปข้อ (5)

(7) เปลี่ยนแปลงภาพบนจอภาพตามชนิดของคำสั่งที่ได้รับมา และปรับแก้ข้อมูล (สถานะของตัวหมากบนกระดานทั้งหมด) ให้ถูกต้อง

(8) เช็คว่าเกมถึงจุดแพ้ชนะกันแล้วหรือยัง ถ้ายังไม่จบก็ให้วนกลับไปที่ข้อ (5)

การที่ผมเรียกโครงสร้างข้างบนนี้ว่า "โครงสร้างพื้นฐาน" ก็เพราะมันจะเป็นโครงสร้างที่เราพบเห็นกันบ่อยที่สุด มันจะไหลลงมาจากข้างบนเพื่อเซตสถานะต่าง ๆ ให้อยู่ในจุดเริ่มต้น เสร็จแล้วมันจะมารอรับคำสั่งของผู้ใช้โปรแกรม ได้รับคำสั่งไปแล้วก็ต้องนำไปตรวจเช็ค (Editing) ถ้าเห็นว่าข้อมูลที่ได้รับเข้ามานั้นอยู่ในสภาพปกติก็จะดำเนินการให้ เมื่อทำแล้วก็จะวนกลับไปรับคำสั่งกันใหม่ โครงสร้างอันนี้สามารถก๊อบปี้ไปใช้ต่อ ๆ กันได้ แต่มันจะมีข้อเสียอยู่ตรงที่ว่า มันยัง

ไม่มีจุดจบ นอกจากนี้แล้ว มันก็ยังไม่มีการเปลี่ยนไปให้ทางด้านคอมพิวเตอร์เดินบ้าง แต่อย่างไรก็ตาม นี่น่าจะเป็นข้อดี คือเราจะได้เช็คแต่ทางด้านการเดินโดยคนเสียก่อน เราจะไม่ต้องไปกังวลกับส่วนที่เป็นลอจิกของการเดินด้วยคอมพิวเตอร์เลย วิธีการนี้ผมเรียกว่า step-by-step Development มันจะทำให้เราทำโปรแกรมได้เร็วขึ้น

หลังจากที่ท่านสามารถทำให้โปรแกรมวิ่งได้ตามโครงสร้างพื้นฐานแล้วก็จะมาเติมส่วนประกอบอื่น ๆ สิ่งแรกที่ท่านต้องเติมเข้าไปก็คือ ให้เพิ่มคำสั่งที่คนจะสั่งได้มากขึ้น เช่น กดแป้นอักษร "E" ให้หมายถึงความต้องการที่จะจบเกมก่อนกำหนด กดแป้นอักษร "X" ให้หมายถึงการสลับข้างกันเล่น กดแป้นอักษร "C" ให้หมายถึงการยกเลิกการมาร์คตัวเดิน กดแป้นอักษร "P" ให้หมายถึงการสั่งให้เอาข้อมูลบางอย่างออกมาแสดงบนจอภาพ ทั้งนี้ก็เพื่อช่วยในการดีบัก เมื่อเติมสิ่งเหล่านี้เข้าไปได้แล้วก็จะมาเติมเรื่องของการให้เครื่องคำนวณและบอกว่าจะเดินตอบได้ออกมาอย่างไร

ในเรื่องของการให้เครื่องคิดและเดินนี้เราควรระวังอยู่อย่างหนึ่ง คือ **ลอจิกของโปรแกรมอาจผิดพลาดบกพร่องได้** ด้วยเหตุนี้เราจึงไม่ควรให้มีการปรับแก้สถานะของข้อมูลบนกระดาน หรือการแก้ไขภาพบนจอภาพโดยลอจิกส่วนนี้โดยเด็ดขาด โปรแกรมส่วนนี้จะเป็นการคำนวณล้วน ๆ ได้ผลเป็นอย่างไรแล้วต้องส่งออกมาให้เหมือนกับการเล่นโดยคน เช่น ต้องบอกว่าจะเดินตัวไหน และจะเดินไปที่จุดไหน หรือถ้าเป็นการเปิดตัวใหม่ ก็จะบอกว่าให้เปิดตัวไหนเท่านั้น ข้อมูลจะส่งผ่านออกมาสู่โปรแกรมที่เป็นโครงสร้างพื้นฐานแล้วจะต้องผ่านขั้นตอนในข้อ (6)...(7)...(8) ตามปกติ ในกรณีที่โปรแกรมส่วนใหม่นี้คำนวณผิดมันจะถูกจับได้โดย (6)...(7)...(8) ดังกล่าว

โดยสรุปแล้วเราจะตัดต่อในบริเวณข้อ (4) และเติมข้อที่จะทำให้โปรแกรมจบลงได้อย่างถูกต้องและครบถ้วน ดังต่อไปนี้

(1) จัดการวาดภาพเป็นกระดานหมากรุกขึ้นมา 1 กระดาน

(2) จัดการหยอดตัวหมากของฝ่ายทั้ง 2 ลงไปอย่าง Randomly ฝ่ายละ 16 ตัว โดยให้อยู่ในสถานะคว่ำหน้า

(3) วิ่งมารับค่าของ score-limit

(4) เซตให้ฝ่าย "คน" เป็นผู้เล่นก่อน

(4x) เช็คว่าฝ่ายไหนจะเป็นผู้เล่น ถ้าฝ่าย "คอมพิวเตอร์" เป็นผู้เล่นก็ให้เรียกโพรซีเดอร์ COMMOVE (....) ออกมาทำงานแล้วกระโดดไปที่ข้อ (6) เลย

(5) วนอยู่ในลูปเพื่อรับคำสั่งจากคน

(5x) ถ้าเป็นอักษร "E" หรือ "X" หรือ "P" หรือ "C" ให้ดำเนินการตามคำสั่ง เสร็จแล้วให้วนกลับไปที่ข้อ (5)

(6) ตรวจคำสั่งการเล่น ถ้าผิดกฎเกณฑ์ให้ดำเนินการดังนี้

(6.1) ถ้าคนเป็นฝ่ายเล่น ให้ทำเสียง "Beep" แล้วกลับไปข้อ (5)

(6.2) ถ้าคอมฯเป็นฝ่ายเล่น ให้ dump ข้อมูลแล้วไปข้อ (10)

(7) เปลี่ยนแปลงภาพบนจอภาพตามคำสั่ง และปรับแก้ข้อมูล (สถานะของตัวหมากบนกระดาน... ฯลฯ) ให้ถูกต้อง

(8) เช็คว่าเกมถึงจุดแพ้ชนะกันแล้วหรือยัง ถ้าถึงจุดแพ้ชนะก็ให้แสดงผลแล้วไปข้อ (10) เพื่อจบโปรแกรม แต่ถ้ายังไม่จบเกมให้ทำข้อถัดไป

(9) เปลี่ยนฝ่ายผู้เล่น แล้วกลับไปข้อ (4x)

(10) จบโปรแกรม

โครงสร้างของโพรซีเดอร์ COMMOVE

COMMOVE (.....) จะเป็นโพรซีเดอร์ที่เล็กก็ได้ ใหญ่ก็ได้ มันจะขึ้นอยู่กับความสนใจของผู้เขียนโปรแกรม ถ้าเราอยากจะให้เจ้าเครื่องคอมพิวเตอร์เล่นเก่ง ๆ เราก็ต้องไปคิดหากลยุทธ์ที่ดี ๆ มาใส่เข้าไป ในกรณีที่ท่านไม่แน่ใจว่ากลยุทธ์อันไหนจะดีกว่าอันไหนก็อาจเขียนขึ้นมาเป็น 2 โพรซีเดอร์ เช่น อาจเรียกว่า COMMOVE1(....) กับ COMMOVE2(.....) เสร็จแล้วไปปรับแก้ในข้อ (4x) เสียหน่อยหนึ่ง คือแทนที่จะเรียกหา COMMOVE(....) ก็ให้เขียนว่า

```
IF CURSIDE = RED THEN COMMOVE1(...)
      ELSE COMMOVE2(...);
```

ด้วยวิธีนี้เราจะพบว่า เวลาที่ฝ่ายแดงเป็นฝ่ายเล่นมันจะเล่นด้วยลอจิกของ COMMOVE1(....) แต่ถ้าฝ่ายขาว (ในเครื่องคอมพิวเตอร์) นั้นเราต้องเปลี่ยนมาใช้สีแดงกับขาว ทั้งนี้เพราะสีดำจะมองไม่เห็น) เป็นผู้เล่น มันก็จะเล่นด้วยลอจิกของ COMMOVE2(....) เมื่อแก้แล้ว เราก็เล่นโดยกดแป้น "X" เพื่อให้มันเปลี่ยนข้างกับตัวท่านไปเรื่อย ๆ ด้วยวิธีนี้จะพบว่าเครื่องจะใช้ลอจิกของ COMMOVE1(....) แข่งกับ COMMOVE2(.....) โดยอัตโนมัติ

นั่นเป็นเกล็ดเล็ก ๆ ที่ได้จากการใช้โครงสร้างของโปรแกรมตามที่ผมได้แนะนำมา แต่ประเด็นตอนนี้ก็คือ ภายในโพรซีเดอร์ COMMOVE(....) นั้นเราควรจะวางโครงสร้างไว้อย่างไร เรื่องนี้ก็ต้องย้อนกลับไปดูในหัวข้อเรื่อง "กลยุทธ์ในการเล่น" ที่ผ่านมา ในโปรแกรมตัวอย่างที่ผมเขียนขึ้นมานั้นจะเป็นการมองภาพเพียงชั้นเดียว นั่นคือ มันจะต้องคำนวณดูว่ามีหนทางเดินหรือเปิดตัวที่ตรงไหนได้บ้าง ส่วนที่ให้เขียนเป็นโพรซีเดอร์ย่อยในชื่อว่า GENMOVES(....) เอาไว้ก่อน แล้วเราค่อยมาเรียกใช้ภายใน COMMOVE(....) อีกต่อหนึ่ง หลังจากที่สร้างเทเบิลที่บอกหนทางเลือกต่าง ๆ แล้วก็มาคำนวณค่าการสูญเสียสูงสุดในแต่ละทางเลือก (=CLST) นี่คือการมองดูว่าฝ่ายตรงข้ามจะโต้ตอบมาได้รุนแรงที่สุดเป็นเท่าใด จากค่าของการได้กิน (=CTK, ถ้ามี) กับค่าของการสูญเสียสูงสุด (=CLST) เราจะต้องมาจัดแยกประเภท ซึ่งผมให้แยกเป็นประเภทต่าง ๆ ดังตารางที่ 1

ตารางที่ 1

TPE 11 = เดินไปกินฝ่ายตรงข้าม 1 ตัว และคาดว่าไม่มีการสูญเสีย
TPE 12 = เดินไปกินฝ่ายตรงข้าม 1 ตัว และอาจถูกกินกลับ 1 ตัว
TPE 13 = เดินไปไม่ได้กิน และคาดว่าไม่มีการสูญเสีย
TPE 90 = เดินไปไม่ได้กิน แล้วยังคาดว่าจะถูกกินกลับ 1 ตัว
TPE 60 = เปิดตัวคว่ำ และคาดว่าไม่มีการสูญเสีย
TPE 80 = เปิดตัวคว่ำ และคาดว่าจะมีการสูญเสีย

จากตารางที่ 1 เราจะเห็นได้ว่า ผมพยายามที่จะให้รหัสมีค่าน้อยแก่ทางเลือกที่ดี และให้รหัสที่มีค่ามากเมื่อเป็นทางเลือกที่เลว และเมื่อคำนวณรหัสชนิดหรือประเภทเสร็จแล้วเราก็จะหาดูว่าในทางเลือกทั้งหมดนี้รหัสต่ำสุดมีค่าเป็นเท่าใด ถัดจากนั้นมาเราก็ต้องมาเลือกเอาจากในหมู่ที่มีค่า TPE ต่ำสุด

โดยสรุปแล้ว โครงสร้างของโพรซีเดอร์ COMMOVE จะเป็นดังนี้

(1) เคลียร์ค่าต่าง ๆ
(2) เรียก GENMOVES มาทำงาน
(3) คำนวณค่า CLST
(4) คำนวณค่า TPE
(5) หาค่าต่ำสุดของ TPE (= MINTPE)
(6) เลือกทางเลือกจากในหมู่ที่ TPE มีค่าเท่ากับ MINTPE

สรุป

ผมอยากจะขอนำเอาสิ่งต่าง ๆ ที่สำคัญ ๆ มาสรุปเอาไว้ตรงนี้เสียหน่อย ซึ่งมีดังนี้

(1) เกมนี้เป็นเกมกระดาน
(2) กระดานมีขนาด 8 x 8 ช่อง เหมือนกระดานหมากรุกไทย
(3) ตัวที่จะใช้เล่นมีฝ่ายละ 16 ตัว แต่ละตัวมี 2 หน้า
(4) เริ่มต้นด้วยการคว่ำหน้าตัวหมากทั้งหมดลงไปบนกระดาน, คนให้ทั่ว, แล้ววางกระจายไปทั่วแผ่นกระดาน
(5) ผู้เล่นเลือกได้ 2 อย่าง คือเดินตัวของตนที่หงายขึ้นมาแล้ว หรือไม่ก็เปิดตัวที่ยังคว่ำหน้าอยู่ขึ้นมา
(6) การเดินจะเดินได้แค่ 4 ทิศ และเป็นระยะทาง 1 ช่อง
(7) เดินไปกินฝ่ายตรงข้ามจะได้คะแนน 3 คะแนน
(8) เปิดได้ตัวของฝ่ายตนจะได้คะแนน 1 คะแนน แต่ถ้าเปิดได้ตัวของฝ่ายตรงข้ามจะเสียคะแนน 2 คะแนน
(9) ตัวเดินจะเดินได้ 9 ครั้งเท่านั้น เมื่อครบ 9 ก็จะถูกยกออก
(10) แพ้-ชนะดูได้ 2 กรณี กรณีแรกคือหมดหนทางเล่น (เดินไม่ได้และไม่มีตัวคว่ำให้เปิด) ส่วนกรณีที่ 2 คือ เมื่อผลต่างของคะแนนมีค่ามากกว่าหรือเท่ากับ score-limit

จากกฎเกณฑ์ต่าง ๆ ที่กล่าวมาข้างต้นนี้ท่านผู้อ่านสามารถนำไปคิดและเขียนออกมาเป็นโปรแกรมได้ และผมเสนอแนะให้ท่านลองใช้ความสามารถของท่านดู โจทย์อันนี้คงจะไม่ยากเกินไปนัก ผู้มีความชำนาญในการออกแบบและเขียนโปรแกรมคงจะทำได้ใน 5 - 6 วัน ส่วนท่านที่ไม่มีพื้นฐานมาก่อนเลยนั้นก็อาจต้องใช้เวลาสัก 20 - 30 วัน แต่อย่างไรก็ตาม ผมได้จัดทำเป็นซอฟต์เมล์ไว้แล้ว เขียนด้วยเทอร์โบปาสคาล เวอร์ชัน 3.0 ตัวซอร์สโค้ดชื่อ CHNS01.PAS ตัวออบเจ็กต์โค้ดชื่อ CHNS01.COM ทั้งหมดนี้ต้องรันภายใต้ดอส เวอร์ชัน 3.1 ขึ้นไป เครื่องจะต้องมีเมมโมรีไม่น้อยกว่า 64 กิโลไบต์ (มันเล็กจนไม่น่าจะเรียกว่าเป็นข้อกำหนด) และใช้จอภาพสีชนิด CGA ขึ้นไป (หลายท่านคงจะลืมไปแล้วว่ามาตรฐาน CGA คืออะไร) ถ้าท่านคิดไม่ออกหรืออยากจะใช้วิธีเรียนลัด ก็สามารถติดต่อซอฟต์เมล์ไปลองเล่น หรือนำไปศึกษาดูได้

ถ้า... ทำทุกอย่างแล้วก็ยังไม่ค่อยจะเข้าใจ ผมก็ขอให้รออ่านฉบับหน้า ในตอนที่ 2 นั้นผมจะพูดในรายละเอียดว่าโปรแกรมมีตัวแปรชื่ออะไรบ้าง และการคำนวณแต่ละขั้นตอนมันทำกันได้อย่างไร

อนึ่ง สำหรับคนที่ไม่มีเครื่องคอมพิวเตอร์ หรือมีแต่ไม่อยากเสียเวลา ท่านก็อาจลองตัดกระดาษเป็นชิ้นกลม ๆ เท่าเหรียญสิบมาใช้เป็นตัวหมาก แล้วไปขอยืมแผ่นกระดานหมากรุกลงลองเล่นกันดูได้ เกมนี้เข้าใจง่ายกว่าเกม "โก๊ะ" ที่คนญี่ปุ่นชอบเล่นเสียอีก